

Compte rendu : Création d'un portfolio

Cet atelier m'a permis de préparer toutes les ressources nécessaires (CV, photo, captures de projets), de me connecter à la plateforme **DeepSeek** et de générer un premier code HTML pour mon portfolio. J'ai aussi appris à organiser correctement les fichiers localement pour faciliter la personnalisation et le futur hébergement.

Atelier 01-1

Exercice 1 : Accès à DeepSeek et préparation des ressources clés

- **Connexion à DeepSeek** : Je me suis rendu sur le site *DeepSeek.com*, puis je me suis connecté avec mon compte.
- **Préparation du CV** : J'ai optimisé et mis à jour mon CV en format PDF. Il contient mes coordonnées, mon profil, mes formations, mes projets, mes expériences professionnelles ainsi que mes liens vers LinkedIn et GitHub (si j'en ai)
- **Photo de profil** : J'ai choisi une photo professionnelle, renommée simplement en *my photo.jpg* pour éviter tout problème d'intégration dans le code.
- **Captures de projets (optionnel)** : J'ai préparé quelques screenshots de mes projets afin de ne pas utiliser uniquement des images de substitution.

☑ Validation :

- Je suis bien connecté à DeepSeek.
- Mon CV est prêt et complet avec mes liens sociaux.
- Ma photo est correctement renommée et prête à l'emploi.
- J'ai des captures de projets disponibles pour personnaliser le portfolio.

Exercice 2 : Téléchargement des fichiers sur DeepSeek et soumission du prompt

- J'ai téléversé mon **CV en PDF** sur la plateforme.
- J'ai téléversé ma **photo de profil (my photo.jpg)** sur DeepSeek.

- J'ai rédigé et soumis un **prompt précis** demandant à DeepSeek de générer un **portfolio complet** en HTML, CSS et JavaScript avec Bootstrap
 - Ce prompt incluait la structure des sections (Accueil, À propos, Compétences, Projets, Expérience, Contact).
 - J'ai demandé un design professionnel en français, avec des visuels attrayants, l'intégration de ma photo et un bouton permettant de télécharger directement mon CV.

✓ Validation :

- Mon CV et ma photo sont bien téléversés.
- Le prompt a été soumis de façon claire et détaillée.

Exercice 3 : Génération du code HTML et organisation locale des fichiers

- **Observation du code généré** : DeepSeek a produit un fichier HTML complet intégrant CSS et JavaScript. J'ai pris le temps de lire et comprendre la structure (balises principales, sections, styles intégrés).
- **Organisation locale** : J'ai compris qu'il est indispensable de réunir dans un même dossier local :
 - Le fichier index.html (mon portfolio),
 - Ma photo de profil,
 - Mon CV en PDF,
 - Mes éventuelles captures de projets.

Cela garantit que mon portfolio fonctionne correctement en local et pourra être facilement déployé par la suite.

✓ Validation :

- Le code HTML complet du portfolio a bien été généré.
- J'ai compris l'importance de l'organisation locale des fichiers (HTML + ressources associées dans un seul dossier).

Conclusion

Grâce à cet atelier, j'ai :

- Préparé et optimisé mes ressources personnelles (CV, photo, captures).
- Téléversé ces ressources sur DeepSeek.
- Rédigé un prompt clair pour générer un portfolio complet et professionnel.

- Analysé le code HTML produit et compris comment organiser mes fichiers localement pour un bon fonctionnement.

Atelier 01-2

1. Formulation du Prompt sur DeepSeek

- J'ai rédigé une instruction claire et détaillée pour demander à DeepSeek de générer le code **HTML, CSS et JavaScript** de mon portfolio.
- J'ai précisé que mon **CV** et ma **photo** avaient déjà été téléversés, afin qu'ils soient intégrés.
- J'ai insisté sur l'utilisation des informations de mon CV (adresse, description, parcours scolaire, projets, expériences et liens LinkedIn/GitHub).
- Ensuite, j'ai soumis le prompt et j'ai obtenu plusieurs blocs de code générés par DeepSeek.

✓ Validation : Mon prompt a bien été pris en compte et le code du portfolio a été généré.

2. Assemblage et Sauvegarde du Code

- J'ai copié soigneusement toutes les parties du code proposées par DeepSeek.
- J'ai ouvert **VS Code**, créé un fichier, puis collé le code complet à l'intérieur.
- J'ai sauvegardé ce fichier sous le nom **index.html**, dans le même dossier que mon CV et ma photo de profil.

✓ Validation : Le fichier **index.html** contient tout le code et se trouve dans le dossier de mon projet.

3. Visualisation Locale du Portfolio

- J'ai ouvert **index.html** dans VS Code.
- Avec l'extension **Live Server**, j'ai lancé un aperçu dans mon navigateur (Chrome).
- Mon portfolio s'est affiché correctement en local.

✓ Validation : Le portfolio s'ouvre bien dans le navigateur.

4. Personnalisation des Liens Sociaux

- J'ai recherché les balises `<a>` dans le code et remplacé les `href="#"` par mes vrais liens **Instagram** (seul réseau social que j'utilise)
- J'ai aussi vérifié que les liens fonctionnaient en cliquant dessus depuis le navigateur.

✓ Validation : Tous mes liens sociaux redirigent vers mes profils personnels.

5. Configuration de la Photo de Profil

- J'ai identifié la balise `` correspondant à ma photo.
- J'ai vérifié que l'attribut `src` correspondait exactement au nom de mon fichier (ex. `ma_photo.jpg`).
- Comme ma photo est bien dans le dossier du projet, elle s'affiche correctement.

✓ Validation : Ma photo de profil apparaît bien sur le portfolio.

6. Configuration du Bouton de Téléchargement du CV

- J'ai trouvé la balise <a> du bouton **"Télécharger mon CV"**.
- J'ai modifié l'attribut « href » pour qu'il pointe vers le nom exact de mon fichier CV (ex. mon_cv.pdf).
- Après enregistrement, j'ai testé le bouton : le téléchargement démarre correctement.

☑ Validation : Le bouton permet bien de télécharger mon CV.

7. Personnalisation du Contenu et des Images (Optionnel)

- J'ai modifié certains textes générés automatiquement (nom, description, titres de sections) pour qu'ils reflètent mieux mon profil.
- J'ai prévu d'ajouter mes propres captures de projets plus tard, en remplaçant les images génériques par mes screenshots .

☑ Validation : Le contenu textuel correspond désormais à mes informations personnelles.

Conclusion et Transition

À l'issue de cet atelier, j'ai généré, assemblé et personnalisé mon portfolio avec **DeepSeek**. Mon CV, ma photo et mes liens sociaux sont intégrés et fonctionnels. Le site est déjà consultable en local.

👉 La prochaine étape sera de préparer son **hébergement en ligne** pour le rendre accessible publiquement (Partie 3).

Atelier 01-3

Après avoir généré et personnalisé mon portfolio en local, j'ai appris à le rendre **accessible en ligne**. Pour cela, j'ai utilisé **GitHub** (stockage du code) et **Netlify** (déploiement automatique). Mon objectif était de permettre à n'importe qui de consulter mon travail via un lien partagé.

Exercice 1 : Comprendre le besoin d'hébergement

- J'ai compris que lorsque j'ouvrais mon fichier **index.html** en local (via *Live Server* ou double-clic), le site était accessible **uniquement sur mon ordinateur**.
- L'URL commençait par *localhost* ou par un chemin de fichier local.
- Pour le rendre visible aux recruteurs et collaborateurs, j'ai vu qu'il fallait **l'héberger sur un serveur distant**.

✓ Validation : J'ai bien compris la différence entre un site local et un site hébergé.

Exercice 2 : Création des comptes GitHub et Netlify

- J'ai créé un compte sur **GitHub** en m'inscrivant avec mon adresse mail et en choisissant un nom d'utilisateur.
- J'ai ensuite créé un compte sur **Netlify**. Je me suis inscrit directement avec GitHub, ce qui a facilité la création de compte.
- J'ai autorisé Netlify à accéder à mon compte GitHub.

✓ Validation : J'ai un compte GitHub fonctionnel et un compte Netlify lié à GitHub.

Exercice 3 : Création du dépôt GitHub et téléversement du projet

- J'ai créé un **nouveau repository public** sur GitHub que j'ai nommé *mon-portfolio*.
- Ensuite, j'ai téléversé tous mes fichiers :
 - **index.html**
 - Ma **photo de profil** (ex. *ma_photo.jpg*)
 - Mon **CV en PDF**
 - Les images de mes projets

- J'ai ajouté un message de commit (*Initial commit - Upload portfolio files*) et validé l'ajout.

✓ Validation : Tous mes fichiers sont bien téléversés dans un dépôt GitHub public.

Exercice 4 : Déploiement du portfolio sur Netlify

- Dans Netlify, j'ai choisi "**Add new site**" → "**Import an existing project**".
- J'ai sélectionné **GitHub** et autorisé l'accès à mes dépôts.
- J'ai choisi mon dépôt *mon-portfolio*.
- Netlify a détecté automatiquement la configuration et j'ai lancé le déploiement.
- Quelques minutes après, Netlify m'a donné une **URL unique** pour accéder à mon site.

✓ Validation : Mon portfolio est désormais **accessible en ligne via une URL Netlify**.

Exercice 5 : Vérification et personnalisation post-déploiement

- J'ai ouvert l'URL Netlify et vérifié que :
 - Ma **photo de profil** s'affichait correctement.
 - Le bouton "**Télécharger mon CV**" fonctionnait.
- J'ai appris que je pouvais personnaliser le nom de domaine Netlify (ex. passer de *nom-aléatoire.netlify.app* à *mon-nom-portfolio.netlify.app*).
- J'ai aussi testé une **mise à jour** :
 - J'ai modifié du texte directement dans **index.html** sur GitHub.
 - J'ai fait un nouveau commit.
 - Netlify a automatiquement redéployé mon site avec la modification.

✓ Validation : Mon portfolio est en ligne, fonctionnel et je sais maintenant le mettre à jour via GitHub.

Conclusion et prochaine étape

Grâce à cet atelier, j'ai :

- Hébergé mon portfolio sur **GitHub** et **Netlify**.
- Vérifié le bon fonctionnement des éléments (photo, CV, liens sociaux).

- Appris à modifier mon site depuis GitHub avec redéploiement automatique sur Netlify.

👉 La prochaine étape sera d'apprendre à utiliser les **commandes Git** (*git init, git add, git commit, git push*) pour une gestion de version plus professionnelle et des déploiements encore plus efficaces.

Atelier 01-4

Après avoir hébergé mon portfolio via GitHub et Netlify avec un simple téléversement de fichiers, j'ai appris à utiliser **Git en ligne de commande**. Git est l'outil de gestion de version local, et GitHub est la plateforme distante qui héberge mes dépôts. Cet atelier m'a permis de comprendre la différence entre les deux et de pratiquer les **commandes Git essentielles** pour gérer et synchroniser mon projet.

Exercice 1 : Comprendre Git et GitHub

- J'ai compris que **Git** est installé et utilisé localement pour suivre l'évolution de mon code.
- **GitHub**, lui, sert à héberger mes dépôts Git sur le cloud et à collaborer.
- Pour interagir de manière professionnelle avec GitHub, je dois utiliser Git en ligne de commande.

✅ Validation : Je distingue bien Git (local) et GitHub (distant).

Exercice 2 : Installation de Git sur ma machine

- J'ai téléchargé Git depuis le site officiel (*git-scm.com/downloads*).
- J'ai lancé l'installation avec les paramètres par défaut.
- Une fois installé, j'ai ouvert **Git Bash** et tapé `git --version` pour vérifier.

✅ Validation : Git est installé et fonctionnel sur mon ordinateur.

Exercice 3 : Création d'un dépôt GitHub

- Je me suis connecté à GitHub.
- J'ai créé un dépôt public vide appelé **mon-portfolio-git-commands**.
- GitHub m'a affiché une série de commandes Git à utiliser pour connecter mon projet.

✓ Validation : Le dépôt public vide est bien créé sur GitHub.

Exercice 4 : Initialisation de Git dans mon dossier local

- J'ai ouvert mon terminal et navigué jusqu'au dossier de mon portfolio (cd puis ls pour vérifier).
- J'ai exécuté git init pour transformer mon dossier en dépôt Git.
- Cela a créé un dossier caché **.git** qui contient l'historique local.

✓ Validation : Mon projet local est désormais un dépôt Git.

Exercice 5 : Ajout et commit des fichiers

- J'ai ajouté tous mes fichiers avec la commande git add ...
- Ensuite, j'ai enregistré mon premier commit avec :
git commit -m "Initial commit - ajout du portfolio"
- Git m'a demandé de configurer mon identité (git config user.name et git config user.email).
- J'ai ensuite pu valider mes modifications.

✓ Validation : Mon premier commit a été créé dans l'historique Git local.

Exercice 6 : Configuration de la branche principale et liaison au dépôt distant

J'ai renommé la branche principale en **main** avec :

```
git branch -M main
```

J'ai lié mon dépôt local au dépôt GitHub créé plus tôt avec :

```
git remote add origin https://github.com/mon\_compte/mon-portfolio-git-commands.git
```

J'ai préparé l'envoi avec :

```
git push -u origin main
```

✓ Validation : Mon projet local est maintenant connecté à mon dépôt GitHub.

Exercice 7 : Push des modifications vers GitHub

- J'ai exécuté `git push -u origin main` pour envoyer mon commit local sur GitHub.
- Git m'a demandé de m'authentifier via mon navigateur (compte GitHub).
- Après validation, mes fichiers ont été envoyés avec succès.

✓ Validation : Mes fichiers et mon historique de commits sont visibles sur GitHub.

Exercice 8 : Vérification des changements sur GitHub

- J'ai actualisé la page de mon dépôt GitHub et j'ai vu :
 - Tous mes fichiers du portfolio.
 - Mon message de commit.
- Comme mon dépôt est lié à **Netlify**, le site a été automatiquement redéployé après mon push.

✓ Validation : J'ai confirmé que mes changements locaux apparaissent bien sur GitHub et se répercutent automatiquement en ligne via Netlify.

Conclusion

Cet atelier m'a permis de :

- Installer et utiliser **Git en ligne de commande**.
- Comprendre le rôle de Git (local) et GitHub (distant).
- Initialiser un dépôt local, y ajouter des commits, le connecter à un dépôt distant et pousser les modifications.
- Vérifier que mes changements se synchronisent avec **GitHub** et se redéployent automatiquement sur **Netlify**.